

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-298717

(43)Date of publication of application : 29.10.1999

(51)Int.Cl.

H04N 1/387

B41J 21/00

G06F 3/12

G06T 1/00

(21)Application number : 10-117851

(71)Applicant : OKI DATA CORP

(22)Date of filing : 13.04.1998

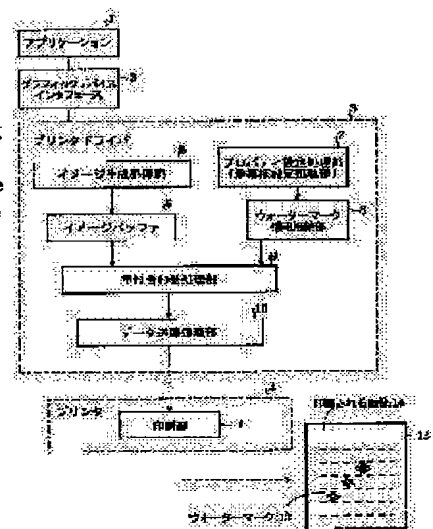
(72)Inventor : ISHIZAKI KOJI

## (54) PRINTING SYSTEM

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To reduce the loads imposed on a printer and to accelerate a printing speed as well by superimposing and compositing a watermark and an image to be printed before sending the data to the printer.

**SOLUTION:** A printer driver 3 generates the bit map data of a watermark in a property setting processing part 7. Also, in an image generation processing part 5, the bit map data of the image to be printed are generated. A superimposing processing part 9 superimposes the image to be printed on the watermark, and thereafter the data are sent to the printing part 11 of the printer 4 through a data transmission processing part 10.



## LEGAL STATUS

[Date of request for examination]

31.01.2002

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

3680109

[Date of registration]

27.05.2005

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-298717

(43) 公開日 平成11年(1999)10月29日

(51) Int.Cl.<sup>a</sup>

識別記号

F I

H 0 4 N 1/387

H 0 4 N 1/387

B 4 1 J 21/00

B 4 1 J 21/00

Z

G 0 6 F 3/12

G 0 6 F 3/12

F

H

G 0 6 T 1/00

15/66

4 5 0

審査請求 未請求 請求項の数11 F D (全 14 頁)

(21) 出願番号

特願平10-117851

(22) 出願日

平成10年(1998)4月13日

(71) 出願人 591044164

株式会社沖データ

東京都港区芝浦四丁目11番地22号

(72) 発明者 石崎 浩司

東京都港区芝浦四丁目11番地22号 株式会  
社沖データ内

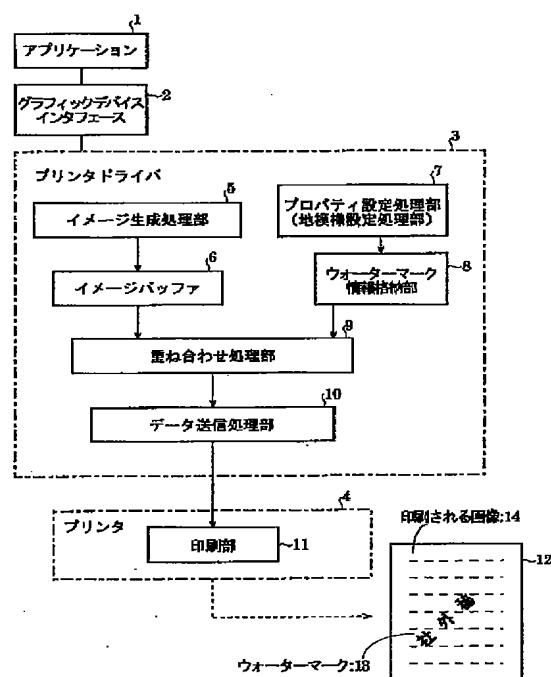
(74) 代理人 弁理士 佐藤 幸男 (外1名)

(54) 【発明の名称】 印刷システム

(57) 【要約】

【解決手段】 プリンタドライバ3は、プロパティ設定処理部7においてウォーターマークのビットマップデータを生成する。また、イメージ生成処理部5において、印刷すべき画像のビットマップデータを生成する。重ね合わせ処理部9は、ウォーターマークに印刷すべき画像を重ね合わせ、その後、データ送信処理部10を通じてプリンタ4の印刷部11にデータが送られる。

【効果】 プリンタにデータを送る前にウォーターマークと印刷すべき画像とを重ね合わせ、合成してしまうので、プリンタの負荷が減り、印刷速度も向上する。



本発明による印刷システムのブロック図

**【特許請求の範囲】**

**【請求項1】** 用紙上に予め印刷されるべき地模様のビットマップデータを指定する地模様設定処理部と、前記地模様の印刷された用紙上に、前記地模様に重ねて印刷されるべき任意の画像のビットマップデータを生成するイメージ生成処理部と、前記地模様設定処理部の出力する地模様のビットマップデータと、前記イメージ生成処理部の出力する画像のビットマップデータとを重ね合わせて印刷部に供給する重ね合わせ処理部とを備えたことを特徴とする印刷システム。

**【請求項2】** プリンタに印刷用データを供給するプリンタドライバ中に、用紙上に予め印刷されるべき地模様のビットマップデータを指定する地模様設定処理部と、前記地模様の印刷された用紙上に、前記地模様に重ねて印刷されるべき任意の画像のビットマップデータを生成するイメージ生成処理部と、前記地模様設定処理部の出力する地模様のビットマップデータと、前記イメージ生成処理部の出力する画像のビットマップデータとを重ね合わせてプリンタに供給する重ね合わせ処理部とを備えたことを特徴とする印刷システム。

**【請求項3】** 請求項2に記載の印刷システムにおいて、地模様設定処理部における地模様の指定は、プリンタドライバのプロパティ設定により行われることを特徴とする印刷システム。

**【請求項4】** 請求項2に記載の印刷システムにおいて、地模様設定処理部は、用紙サイズに対応させて前記地模様のビットマップデータを伸縮処理することを特徴とする印刷システム。

**【請求項5】** 請求項2に記載の印刷システムにおいて、地模様設定処理部は、用紙の指定された印刷領域のサイズに対応させて前記地模様のビットマップデータを伸縮処理することを特徴とする印刷システム。

**【請求項6】** 請求項2に記載の印刷システムにおいて、重ね合わせ処理部は、地模様のビットマップデータ中の、その上に重なる画像のビットマップデータの輪郭部分を白抜き処理することを特徴とする印刷システム。

**【請求項7】** プリンタに印刷用データを供給するプリンタドライバであって、用紙上に予め印刷されるべき地模様のビットマップデータを指定する地模様設定処理機能と、前記地模様の印刷された用紙上に、前記地模様に重ねて

印刷されるべき任意の画像のビットマップデータを生成するイメージ生成処理機能と、前記地模様設定処理部の出力する地模様のビットマップデータと、前記イメージ生成処理部の出力する画像のビットマップデータとを重ね合わせてプリンタに供給する重ね合わせ処理機能とを備えたコンピュータプログラムを記録した、コンピュータ読み取り可能な記録媒体。

**【請求項8】** ユーザーモードで動作するアプリケーションと、このアプリケーションにより起動されてユーザーモードで動作するプリンタユーザーインタフェースドライバと、カーネルモードで動作するプリンタグラフィックドライバとを備え、前記アプリケーションは、任意の文書から成る印刷文書データを作成し、前記プリンタユーザーインタフェースドライバは、前記印刷文書データの印刷要求に伴って設定されたドライバプロパティに基づいて、前記印刷文書データと重ね合わせて印刷をされる地模様を描画するためのデバイス情報を作成し、前記プリンタグラフィックドライバは、前記アプリケーションの印刷要求に伴って入力する関数群を解釈して前記印刷文書データに対応する文書用ビットマップデータを生成するとともに、前記関数群に含まれる所定の呼び出し処理関数により、前記ユーザーインタフェースドライバが前記デバイス情報に基づいて生成した地模様に対応するビットマップデータをバッファに受け入れて格納し、さらに、このバッファに格納したビットマップデータと前記文書用ビットマップデータとを重ね合わせ処理してプリンタに供給することを特徴とする印刷システム。

**【請求項9】** 請求項8に記載の印刷システムにおいて、プリンタグラフィックドライバは、アプリケーションの印刷要求に伴って入力する関数群に含まれる、印刷イベントをフックする関数により、ユーザーインタフェースドライバを呼び出し、ユーザーインタフェースドライバは、アプリケーションプログラムインタフェースによりデバイス情報から地模様のビットマップデータを生成し、プリンタグラフィックドライバは、ユーザーインタフェースドライバからの要求に従って、前記地模様に対応するビットマップデータを格納するバッファを用意し、ユーザーインタフェースドライバは、前記地模様のビットマップデータを前記バッファへ出力し、前記プリンタグラフィックドライバは、前記ユーザーインタフェースドライバの要求により、前記地模様のビットマップデータと前記文書用ビットマップデータとを論

理と処理により重ね合わせて、プリンタに供給することを特徴とする印刷システム。

【請求項10】 請求項8に記載の印刷システムにおいて、ユーザーインタフェースドライバは、地模様を描画する場合にその回転角が指定されたとき、デバイス情報から、前記回転角に応じた自模様の描画領域と描画開始位置とを算出して、生成したビットマップデータを順次プリンタグラフィックドライバに向けて出力することを特徴とする印刷システム。

【請求項11】 ユーザーモードで動作するアプリケーションと、このアプリケーションにより起動されてユーザーモードで動作するプリンタユーザーインタフェースドライバと、カーネルモードで動作するプリンタグラフィックドライバとを備え、前記アプリケーションは、任意の文書から成る印刷文書データを作成し、前記プリンタユーザーインタフェースドライバは、前記印刷文書データの印刷要求に伴って設定されたドライバプロパティに基づいて、前記印刷文書データと重ね合わせて印刷をされる地模様を描画するためのデバイス情報を作成し、前記プリンタグラフィックドライバは、前記アプリケーションの印刷要求に伴って入力する関数群を解釈して前記印刷文書データに対応する文書用ビットマップデータを生成するとともに、前記関数群に含まれる所定の呼び出し処理関数により、前記ユーザーインタフェースドライバが前記デバイス情報に基づいて生成した地模様に対応するビットマップデータをバッファに受け入れて格納し、さらに、このバッファに格納したビットマップデータと前記文書用ビットマップデータとを重ね合わせ処理してプリンタに供給する機能を持つコンピュータプログラムを記録した、コンピュータ読み取り可能な記録媒体。

#### 【発明の詳細な説明】

##### 【0001】

【発明の属する技術分野】 本発明は、用紙上に予め印刷されるべき地模様の上に重ねて文書等の画像を印刷する機能を持つ印刷システムに関する。

##### 【0002】

【従来の技術】 一般に、例えば「社外秘」扱いの文書を配布する場合には、文書の各ページに「社外秘」と表示したスタンプを押す。ところが、ワードプロセッサやコンピュータで印刷した文書を電子メールで送信するような場合に、実際に文書印刷後スタンプを押すといった作業は不可能である。従って、電子化されたデータそのものにこうしたスタンプイメージを印刷するような制御データを組み込む。この「社外秘」といったマークをウオ

ーターマークと呼んでいる。

【0003】 ウォーターマーク付きの文書印刷の場合には、まずアプリケーションがプリンタに対しウォーターマークの印刷を指定するコマンドを送信する。続いて、アプリケーションは印刷すべき文書のビットマップデータをプリンタに送信する。プリンタ側では、内蔵しているフォントを使用してウォーターマークのビットマップデータを生成し、後から送り込まれた文書のビットマップデータと重ね合わせて印刷する。

##### 【0004】

【発明が解決しようとする課題】 ところで、上記のような従来の技術には次のような解決すべき課題があった。ウォーターマーク付きの文書印刷を実行する場合、プリンタは内蔵フォントを展開してウォーターマークのビットマップデータを生成してから印刷を開始する。従って、印刷待ちの時間が長くなるという欠点がある。また、フォントを内蔵していないプリンタでは、このようなウォーターマーク印刷のコマンドには対応できない。従って、ウォーターマークの印刷は無視される。また、ウォーターマークに限らず、用紙に様々なマークや枠や模様（以下、この明細書ではこれらを総称して地模様と呼ぶ）を印刷した上で文書を印刷する要求も少なくない。このような処理を円滑に効率的に行う印刷システムの開発が望まれる。

##### 【0005】

【課題を解決するための手段】 本発明は以上の点を解決するため次の構成を採用する。

〈構成1〉 用紙上に予め印刷されるべき地模様のビットマップデータを指定する地模様設定処理部と、上記地模様の印刷された用紙上に、上記地模様を重ねて印刷されるべき任意の画像のビットマップデータを生成するイメージ生成処理部と、上記地模様設定処理部の出力する地模様のビットマップデータと、上記イメージ生成処理部の出力する画像のビットマップデータとを重ね合わせて印刷部に供給する重ね合わせ処理部とを備えたことを特徴とする印刷システム。

【0006】 〈構成2〉 プリンタに印刷用データを供給するプリンタドライバ中に、用紙上に予め印刷されるべき地模様のビットマップデータを指定する地模様設定処理部と、上記地模様の印刷された用紙上に、上記地模様を重ねて印刷されるべき任意の画像のビットマップデータを生成するイメージ生成処理部と、上記地模様設定処理部の出力する地模様のビットマップデータと、上記イメージ生成処理部の出力する画像のビットマップデータとを重ね合わせてプリンタに供給する重ね合わせ処理部とを備えたことを特徴とする印刷システム。

【0007】 〈構成3〉 構成2に記載の印刷システムにおいて、地模様設定処理部における地模様の指定は、プリンタドライバのプロパティ設定により行われることを特徴とする印刷システム。

【0008】〈構成4〉構成2に記載の印刷システムにおいて、地模様設定処理部は、用紙サイズに対応させて上記地模様のビットマップデータを伸縮処理することを特徴とする印刷システム。

【0009】〈構成5〉構成2に記載の印刷システムにおいて、地模様設定処理部は、用紙の指定された印刷領域のサイズに対応させて上記地模様のビットマップデータを伸縮処理することを特徴とする印刷システム。

【0010】〈構成6〉構成2に記載の印刷システムにおいて、重ね合わせ処理部は、地模様のビットマップデータ中の、その上に重なる画像のビットマップデータの輪郭部分を白抜き処理することを特徴とする印刷システム。

【0011】〈構成7〉プリンタに印刷用データを供給するプリンタドライバであって、用紙上に予め印刷されるべき地模様のビットマップデータを指定する地模様設定処理機能と、上記地模様の印刷された用紙上に、上記地模様に重ねて印刷されるべき任意の画像のビットマップデータを生成するイメージ生成処理機能と、上記地模様設定処理部の出力する地模様のビットマップデータと、上記イメージ生成処理部の出力する画像のビットマップデータとを重ね合わせてプリンタに供給する重ね合わせ処理機能とを備えたコンピュータプログラムを記録した、コンピュータ読み取り可能な記録媒体。

【0012】〈構成8〉ユーザーモードで動作するアプリケーションと、このアプリケーションにより起動されてユーザーモードで動作するプリンタユーザーインタフェースドライバと、カーネルモードで動作するプリンタグラフィックドライバとを備え、上記アプリケーションは、任意の文書から成る印刷文書データを作成し、上記プリンタユーザーインタフェースドライバは、上記印刷文書データの印刷要求に伴って設定されたドライバプロパティに基づいて、上記印刷文書データと重ね合わせて印刷をされる地模様を描画するためのデバイス情報を作成し、上記プリンタグラフィックドライバは、上記アプリケーションの印刷要求に伴って入力する関数群を解釈して上記印刷文書データに対応する文書用ビットマップデータを生成するとともに、上記関数群に含まれる所定の呼び出し処理関数により、上記ユーザーインタフェースドライバが上記デバイス情報に基づいて生成した地模様に対応するビットマップデータをバッファに受け入れて格納し、さらに、このバッファに格納したビットマップデータと上記文書用ビットマップデータとを重ね合わせ処理してプリンタに供給することを特徴とする印刷システム。

【0013】〈構成9〉構成8に記載の印刷システムにおいて、プリンタグラフィックドライバは、アプリケーションの印刷要求に伴って入力する関数群に含まれる、印刷イベントをフックする関数により、ユーザーインタフェースドライバを呼び出し、ユーザーインタフェース

ドライバは、アプリケーションプログラムインタフェースによりデバイス情報から地模様のビットマップデータを生成し、プリンタグラフィックドライバは、ユーザーインタフェースドライバからの要求に従って、上記地模様に対応するビットマップデータを格納するバッファを用意し、ユーザーインタフェースドライバは、上記地模様のビットマップデータを上記バッファへ出力し、上記プリンタグラフィックドライバは、上記ユーザーインタフェースドライバの要求により、上記地模様のビットマップデータと上記文書用ビットマップデータとを論理処理により重ね合わせて、プリンタに供給することを特徴とする印刷システム。

【0014】〈構成10〉構成8に記載の印刷システムにおいて、ユーザーインタフェースドライバは、地模様を描画する場合にその回転角が指定されたとき、デバイス情報から、上記回転角に応じた自模様の描画領域と描画開始位置とを算出して、生成したビットマップデータを順次プリンタグラフィックドライバに向けて出力することを特徴とする印刷システム。

【0015】〈構成11〉ユーザーモードで動作するアプリケーションと、このアプリケーションにより起動されてユーザーモードで動作するプリンタユーザーインタフェースドライバと、カーネルモードで動作するプリンタグラフィックドライバとを備え、上記アプリケーションは、任意の文書から成る印刷文書データを作成し、上記プリンタユーザーインタフェースドライバは、上記印刷文書データの印刷要求に伴って設定されたドライバプロパティに基づいて、上記印刷文書データと重ね合わせて印刷をされる地模様を描画するためのデバイス情報を作成し、上記プリンタグラフィックドライバは、上記アプリケーションの印刷要求に伴って入力する関数群を解釈して上記印刷文書データに対応する文書用ビットマップデータを生成するとともに、上記関数群に含まれる所定の呼び出し処理関数により、上記ユーザーインタフェースドライバが上記デバイス情報に基づいて生成した地模様に対応するビットマップデータをバッファに受け入れて格納し、さらに、このバッファに格納したビットマップデータと上記文書用ビットマップデータとを重ね合わせ処理してプリンタに供給する機能を持つコンピュータプログラムを記録した、コンピュータ読み取り可能な記録媒体。用紙上に予め印刷されるべき地模様のビットマップデータを指定する地模様設定処理部と、上記地模様の印刷された用紙上に、上記地模様に重ねて印刷されるべき任意の画像のビットマップデータを生成するイメージ生成処理部と、上記地模様設定処理部の出力する地模様のビットマップデータと、上記イメージ生成処理部の出力する画像のビットマップデータとを重ね合わせて印刷部に供給する重ね合わせ処理部とを備えたことを特徴とする印刷システム。

【0016】〈構成2〉プリンタに印刷用データを供給

するプリンタドライバ中に、用紙上に予め印刷されるべき地模様のビットマップデータを指定する地模様設定処理部と、上記地模様の印刷された用紙上に、上記地模様に重ねて印刷されるべき任意の画像のビットマップデータを生成するイメージ生成処理部と、上記地模様設定処理部の出力する地模様のビットマップデータと、上記イメージ生成処理部の出力する画像のビットマップデータとを重ね合わせてプリンタに供給する重ね合わせ処理部とを備えたことを特徴とする印刷システム。

【0017】〈構成3〉構成2に記載の印刷システムにおいて、地模様設定処理部における地模様の指定は、プリンタドライバのプロパティ設定により行われることを特徴とする印刷システム。

【0018】〈構成4〉構成2に記載の印刷システムにおいて、地模様設定処理部は、用紙サイズに対応させて上記地模様のビットマップデータを伸縮処理することを特徴とする印刷システム。

【0019】〈構成5〉構成2に記載の印刷システムにおいて、地模様設定処理部は、用紙の指定された印刷領域のサイズに対応させて上記地模様のビットマップデータを伸縮処理することを特徴とする印刷システム。

【0020】〈構成6〉構成2に記載の印刷システムにおいて、重ね合わせ処理部は、地模様のビットマップデータ中の、その上に重なる画像のビットマップデータの輪郭部分を白抜き処理することを特徴とする印刷システム。

#### 【0021】

【発明の実施の形態】以下、本発明の実施の形態を具体例を用いて説明する。

〈具体例1〉図1は、本発明による印刷システムの具体例を示すブロック図である。図のシステムは、上位装置であるパーソナルコンピュータ等でアプリケーション1が作成した文書等の印刷用データを、グラフィックデバイスインタフェース2を通じてプリンタドライバ3に供給し、これをプリンタ4に供給して印刷するよう構成されている。

【0022】アプリケーション1は任意の文書等を作成し、文書データをグラフィックデバイスインタフェース2に供給する。グラフィックデバイスインタフェース2は、グラフィック描画関数を出力デバイスに依存した出力に変換するための、よく知られたモジュールである。プリンタドライバ3は、出力デバイスに依存した印刷データを生成し、それを送信するモジュールである。ここで、出力デバイスであるプリンタ4は、印刷部11において、用紙12に対しウォーターマーク13とこれに重ね合わせて印刷される画像14とを印刷する機能を持つ。

【0023】上記プリンタドライバ3には、イメージ生成処理部5、イメージバッファ6、プロパティ設定処理部7、ウォーターマーク情報格納部8、重ね合わせ処理

部9及びデータ送信処理部10が設けられている。イメージ生成処理部5は、グラフィックデバイスインタフェース2が生成したグラフィック描画コマンドからビットマップデータを生成する部分である。このビットマップデータはイメージバッファ6に一時格納される。

【0024】プロパティ設定処理部7は、印刷に使用する用紙サイズや印刷方向、その他の属性を設定する部分である。なお、本発明では、このプロパティ設定処理部7でウォーターマークの属性を同時に設定する。この具体例では、ウォーターマークと任意の画像との重ね合わせについて説明するが、ウォーターマーク以外の、予め用紙上に下敷きとして印刷される様々な地模様についても、ここで設定を行うようにする。

【0025】プロパティ設定処理部7は、ウォーターマークのプロパティが設定された場合、図示しないフォントメモリや地模様生成のためのデータを蓄積したパターンメモリから、必要なイメージ処理用のデータを取り出して、ビットマップデータを生成する。こうして生成されたウォーターマークのビットマップデータは、ウォーターマーク情報格納部8に一時格納される。

【0026】重ね合わせ処理部9は、イメージバッファ6から読み出した印刷すべき画像のビットマップデータと、ウォーターマーク情報格納部8から読み出したウォーターマークのビットマップデータとを重ね合わせてデータ送信処理部10に送信する処理を行う部分である。データ送信処理部10は、プリンタ4へデータを送信するインタフェースである。

【0027】この具体例では、ウォーターマークを、プリンタの内蔵フォントを使用しないで印刷する。このために、コンピュータ上で、そのグラフィックシステムで提供されるフォントからウォーターマークを生成する。即ち、ウォーターマークのビットマップデータをプリンタ側で印刷時に生成するのではなく、印刷開始前に予めプリンタドライバ3の側で生成しておく。更に、印刷すべき文書等のビットマップデータとウォーターマークとを重ね合わせた状態で、そのデータをプリンタに供給する。

【0028】図2には、プロパティ設定処理部の動作フローチャートを示す。図1に示したアプリケーション1は、プリンタドライバ3のプロパティ設定処理部7を呼び出すことによってウォーターマークの設定を行う。プリンタドライバ3は、例えばパーソナルコンピュータのオペレーティングシステムとプリンタとの間のインタフェースとして動作するプログラムである。即ち、アプリケーション1がプリンタドライバ3に印刷を要求する場合には、プリンタドライバ3に対してプロパティ設定処理のためのメッセージを送り、プリンタドライバ3はこのメッセージを受けて起動する。ここで、プリンタドライバ3は図2に示すような処理を実行する。

【0029】まず、ステップS1において、プロパティ

設定処理を開始すると、例えばパーソナルコンピュータの画面上にプリンタドライバ制御用のウィンドウが表示される。ここで、ウォーターマークの設定を指定するとステップS 2に進み、ウォーターマーク指定画面の表示が行われる。ウォーターマーク指定画面は、この図2の右側に示すような内容となる。

【0030】このプロパティ入力画面16は、例えばウォーターマークのビットマップデータの解像度や、ウォーターマークとして表示する文字列や、用紙に印刷された際のウォーターマークの回転角、濃度等の入力を可能とする。ここにデフォルトで表示された解像度がプリンタの実際の解像度と異なる場合には、プリンタの解像度に合わせてこの部分を変更する。なお、用紙サイズに対応させて、この解像度の設定を変更することによりウォーターマークを伸縮させることもできる。この点は具体例2において更に詳しく説明する。

【0031】文字列としては、ウォーターマークを校正する「社外秘」といった文字を指定する。図2(b)には回転角の説明図を示す。図に示すように、用紙上に一定の角 $\alpha$ の傾きを持ってウォーターマーク13が印刷される場合、この回転角 $\alpha$ を指定する。なお、破線は用紙の主走査方向に平行に引いてあるものとする。

【0032】また、(c)には、濃度の例を示す。ウォーターマークを印刷する場合に、そのビットマップデータを構成する各ドットを、例えばこの図に示すように、 $4 \times 4$ のマトリクスで表現する場合に、マトリクスM1は全面的にこれらのドットを黒に塗りつぶす。この場合を100パーセントの濃度とする。そして、 $4 \times 4$ のマトリクス中、半分のドットだけを塗りつぶすマトリクスM2を50パーセントの濃度とする。こうした塗りつぶし面積の比によってウォーターマークの全体的な濃度を調整し、これと重ね合わせて印刷される文字やその他のデータとの混同を防ぐ。

【0033】図2のステップS 3では、このような操作によってウォーターマークの形式を指定する。そして、ステップS 4で、指定エラーがないと判断されると、ステップS 5に進み、図1に示したプロパティ設定処理部7がウォーターマークのビットマップデータを作成し、ウォーターマーク情報格納部8に格納する。

【0034】図3には、具体例1のプリンタドライバ動作フローチャートを示す。この図を用いて、図1に示すプリンタドライバ3の全体的な動作を説明する。まず、ステップS 1において、グラフィックデバイスインタフェース2からグラフィック描画コマンドを受け取ると、イメージ生成処理部5はそのグラフィック描画コマンドからビットマップデータを生成してイメージバッファ6に格納する。

【0035】次に、ステップS 2において、ウォーターマークを付加するかどうかの判断をする。ウォーターマークを付加する必要がなければステップS 5にジャンプ

する。また、ウォーターマークを付加すると判断した場合にはステップS 3に進み、実際にウォーターマーク情報格納部8にウォーターマークイメージが存在するかどうかを判断する。存在すればステップS 4に進み、ウォーターマークのビットマップデータをウォーターマーク情報格納部8から読み出す。

【0036】そして、イメージバッファ6から印刷すべき画像のビットマップデータを読み出し、重ね合わせ処理部9において、重ね合わせ処理を実行する。重ね合わせ処理部9の動作は、具体的には2つのイメージの論理和演算による重ね合わせ処理による。その後、ステップS 5に進み、データ送信処理部10がプリンタ4の印刷部11に対しそのデータを送信する。なお、ステップS 3において、ウォーターマークのイメージがないと判断されると、ウォーターマークを付加しない扱いとしてそのままステップS 5に進み、通常の印刷が実行される。ウォーターマークを生成しようとしても該当するフォント等が無い場合はこれに該当する。

【0037】〈具体例1の効果〉以上のように、プリンタドライバにおいて、印刷すべきイメージとウォーターマークを重ね合わせた後、プリンタの印刷部に供給するようにしたので、プリンタが印刷直前にウォーターマーク用のフォントを展開するといった処理が不要になり、印刷速度が向上する。また、プリンタ側にフォントが存在しないような場合にもウォーターマークを含む印刷が可能になる。

【0038】〈具体例2〉文書を印刷すべき用紙のサイズが常に一定ではなく、一部に異なるサイズの用紙が含まれているような文書を印刷することがある。こうした場合、既に説明したような処理を行うと、常に一定の大きさのウォーターマークが用紙の上に印刷され、例えば用紙サイズが半分になった場合に、ウォーターマークの一部がはみ出して正常な印刷ができなくなるといった問題も生じる。この具体例は、用紙サイズがまちまちの場合においても、ウォーターマークを良好に印刷できる例を説明する。

【0039】図4に、具体例2のプリンタドライバ動作フローチャートを示す。この図のステップS 1からステップS 3までの処理は、具体例1による図3の動作と同一である。図のステップS 4において、ウォーターマーク自体の基準用紙サイズと現在の用紙サイズとが同じかどうかの判断を行う。これは、ウォーターマークがもともとどのサイズの用紙に適するように調整されているかという点と、アプリケーションで実際の印刷のために指定した用紙サイズやプリンタ側で準備している用紙サイズ等を考慮して判断される。

【0040】また、ウォーターマークの基準用紙サイズと実際の用紙サイズとが一致しない場合にはステップS 4からステップS 7に進み、ウォーターマークのビットマップデータを基準用紙サイズと現在の用紙サイズとの

比率で、拡大あるいは縮小させて、印刷文書のビットマップデータに重ね合わせる。

【0041】例えば、図2に示したように、ウォーターマークの解像度を72dpiとする。即ち、A4判の用紙に72dpiの解像度で印刷すると丁度良い大きさになるようなウォーターマークのビットマップデータが用意されているものとする。これを300dpiでA4判に印刷するとすれば、ドット数が足りないため、縮小されて印刷されてしまう。そこで、実際には5倍にビットマップデータを拡大し、ほぼ300dpiの大きさのウォーターマークを印刷できるようにする。

【0042】ウォーターマーク基準用紙サイズと現在の用紙サイズとが同じ場合、ステップS5に進み、その後ステップS6に進む。このステップS5、S6の処理は、具体例1の場合と同様であり、重複する説明を省略する。

【0043】図5には、ウォーターマークの伸縮処理動作説明図を示す。この図を用いてウォーターマークの別の伸縮処理について説明を行う。図5(a)は、例えば基準用紙サイズが横方向にX1、縦方向にY1である用紙12であるとする。ここにウォーターマーク13が所定の大きさと印刷される。ところが、例えば図5(b)に示すように、印刷をしようとする現在の用紙サイズが横X2、縦Y2であるとする。これに、もし(a)と同一の大きさとウォーターマーク13が印刷されるとすれば現在の用紙からはみ出してしまふ。

【0044】そこで、(c)に示すような縮小処理を実行する。実際には、(c)に示す左側の元の大きさのウォーターマーク13に対しX2/X1の縮小倍率を掛ける。これによって、大きさを変更して現在の用紙サイズに整合させる。

【0045】なお、例えば元のウォーターマーク13のフォントサイズP1、Q1と、実際に印刷される用紙を考慮したフォントサイズP2、Q2がわかっているような場合に、これらの比を考慮して縮小することも可能である。なお、この場合に、ウォーターマークの文書中の傾き等を考慮した縮小計算を行うと、より正確な縮小が可能となる。

【0046】更に、ウォーターマークを印刷することができる領域が指定されていることがある。これは、用紙上の一部の領域である場合もあるし、例えば用紙の余白部分を除外した中央部分であることもある。このような場合、そうした印刷領域のサイズを、例えば図2に示すウォーターマークのプロパティ設定用画面で設定できれば、ウォーターマークの拡大縮小処理が確実にできる。

【0047】〈具体例2の効果〉上記のように、プリンタドライバ側で用紙サイズとウォーターマークのサイズとを考慮して、用紙サイズに対応させたり、あるいは印刷領域のサイズに対応させたウォーターマークのビットマップデータを生成し処理するので、用紙の大きさが切

り替わったり、様々な状況の変化があっても、それに応じて適切な大きさのウォーターマークの印刷が可能となる。

【0048】〈具体例3〉ウォーターマークの上には直接印刷すべき文書の文字等が重ね合わされる。従って、場合によっては、両方のイメージが干渉し合って印刷結果を見にくくしてしまうことがある。この具体例では、こうした干渉を除く例を説明する。具体的には、図1に示した重ね合わせ処理部9が、ウォーターマークのビットマップデータ中の、その上に重なる画像のビットマップデータの輪郭部分を白抜き処理する。即ち、ウォーターマーク上に文字や図形が重なる場合、文字や図形の輪郭に相当する部分について、ウォーターマークを白抜き処理する。従って、文字とウォーターマークとの境界が明らかになり、文字等が読みやすくなる。

【0049】図6には、こうした白抜き処理の動作説明図を示す。まず、図6において、印刷すべき画像中の文字パターン、例えば「の」という字が地模様となるウォーターマーク上に重なるものとする。このとき、図の(b)に示すように、この文字パターンをX方向とY方向とに2ビットずつシフトさせて、「の」を表示するためのドットがトレースした領域を求める。

【0050】これは、図の(b)中の白ドットと黒ドットとを含めたほぼ菱形の領域となる。そして、この領域をカバーするマスクパターンを図の(c)に示すように生成する。図の(b)に示した全てのドットの部分を黒ドットとし、その周りに存在し、一定の四角形の領域に含まれる部分を白ドットとしたものをマスクパターンとした。

【0051】次に、地模様であるウォーターマークを取り出し、図の(d)に示すように地模様パターンとする。そして、(c)に示すマスクパターンと(d)で示すマスクパターンとのNAND(論理積の否定)をとる。これによって、図の(e)に示すような白抜きパターンが形成される。この白抜きパターンと(a)に示す文字パターンとの論理和をとると、図の(f)に示すような出力が得られる。

【0052】〈具体例3の効果〉以上によって、「の」という文字の輪郭部分が白抜きされるため、ウォーターマークの上に重ね合わされる文字が読みづらくなるといったことを防止できる。なお、上記のプロパティ設定処理部7は、ウォーターマークのプロパティが設定されると、必要に応じて適当なフォントデータその他を用いて、ウォーターマークのビットマップデータを生成した。しかしながら、例えば予めウォーターマークが一般のフォントデータと同様にウォーターマークを表示するイメージデータのみから成り、ビットマップデータに展開する場合には、予め指定された印刷位置にそのデータを配置するだけでよいようにすれば、プロパティ設定処理部7の負荷が十分に軽減される。



【0053】また、ウォーターマークをフォントデータのようにして保存すると、用紙全体をカバーするようなウォーターマークのイメージを保存しておく場合よりも十分に少ないメモリ容量でこれが実現できるため、資源の有効利用が図られる。更に、プロパティ設定処理部7によるウォーターマークの生成処理が高速化され、印刷開始前のプリンタドライバ3によるデータの重ね合わせ処理等も高速化される。

【0054】なお、プロパティ設定処理部と、イメージ生成処理部と、重ね合わせ処理部とを、上記各具体例では上位装置側のプリンタドライバに設けた例を説明した。しかしながら、これらは、画データを供給する上位装置からプリンタまでの間のどの部分に設けてもよい。また、重ね合わせ処理部のみプリンタ側に設けるといった構成も可能である。本発明は、ウォーターマークのようなフォントを持たないプリンタにより上記のような文書の印刷を行うのに適するが、フォントを内蔵するプリンタの制御に使用しても構わない。

【0055】さらに、上記の各具体例では、全てウォーターマークを例にとりて説明したが、ウォーターマーク以外に、用紙上に予め印刷されるべき様々な地模様についても同様の処理を行うことができる。

【0056】〈具体例4〉上記のようなウォーターマークの処理をマイクロソフト社によるオペレーティングシステム(WindowsNT4.0)のGDI(Graphics Device Interface)に実行させる場合には次のような問題がある。マイクロソフト社によるオペレーティングシステム(Windows3.1)のプリンタドライバに相当するものは、WindowsNT4.0ではGDIとプリンタグラフィックドライバである。このWindowsNT4.0では、GDIとプリンタグラフィックドライバはカーネルモードで動作する。

【0057】従って、GDIに対してウォーターマークのような描画データを作成する機能を実装するのは難しい。同様に、プリンタグラフィックドライバ自身が描画データを作成するような機能を実装するのも難しい。ユーザーが自由にアプリケーションを組むことができるのは、ユーザーモード部分に限られるからである。また、文書データとウォーターマークとを共にユーザーモードで生成して、これらをファイルに保存して印刷データを作成する方法も考えられる。ところが、カーネルモードで動作するプリンタグラフィックドライバは、ファイルを開く機能を持たないので、これも実現不可能である。そこで、以下の具体例のような構成にする。

【0058】図7に、具体例4の印刷システムブロック図を示す。図の太線20の上に示したブロックはユーザーモードで動作する部分である。アプリケーション1は任意の文書を作成して印刷要求を行うためのプログラムである。プリンタユーザーインタフェースドライバ21は、オペレーティングシステムがユーザーに提供している、プリンタの印刷条件や印刷文書設定のためのダイヤ

ログ表示等を実行するドライバである。

【0059】ここには、用紙の種類、印刷ページ、フォント等、様々な印刷条件を設定するプロパティ設定処理部22と、この発明で必要なウォーターマークのビットマップデータを作成するウォーターマーク作成処理部23が設けられている。アプリケーション1が作成した印刷文書データは、印刷文書メモリ24に一時格納される。

【0060】また、プリンタユーザーインタフェースドライバ21が、その印刷文書データの印刷要求に伴って設定したドライバプロパティに基づいて、印刷文書データと重ね合わせて印刷されるウォーターマーク等の地模様を描画するデータは、デバイス情報としてデバイス情報メモリ25に一時格納される。このデバイス情報は、ウォーターマークのビットマップデータを生成するために必要なコードデータ群等により構成される。

【0061】GDI32.DLL, WinSpool.DLLモジュール26は、アプリケーションが印刷文書を印刷するために呼び出すプログラムモジュールである。このモジュールは、出力デバイスに依存しないグラフィック描画関数を提供する。カーネルモードで動作する部分には、グラフィックデバイスインタフェース30、プリンタグラフィックドライバ31が設けられ、これによって生成された印刷データがプリンタ4に供給される構成になっている。

【0062】グラフィックデバイスインタフェース30は、スプーラ26から入力した描画関数を受け入れて、出力デバイスに依存した印刷データを生成する部分である。プリンタグラフィックドライバ31は、イメージ生成処理部32と、イメージバッファ33と、ウォーターマーク重ね合わせ処理部34と、データ送信処理部35とを備えている。

【0063】イメージ生成処理部32は、グラフィックデバイスインタフェース30から入力する印刷データに基づいて、ビットマップデータを生成する部分である。イメージバッファ33は、後で説明するようにしてプリンタユーザーインタフェースドライバ21から転送されたウォーターマークのビットマップデータを一時格納する部分である。

【0064】ウォーターマーク重ね合わせ処理部34は、イメージ生成処理部32の出力とイメージバッファ33に格納されたウォーターマークとを重ね合わせ処理する部分である。データ送信処理部35は、ウォーターマーク重ね合わせ処理部34の出力をプリンタ4に向けて転送する動作を行う部分である。

【0065】上記のような構成の印刷システムも、具体例1の場合と同様、ウォーターマークのような地模様を、プリンタの内蔵フォントを使用しないで、上位装置であるコンピュータ上で生成する。このウォーターマークの生成は、カーネルモードで動作するプリンタグラフィックドライバ31ではなく、ユーザーモードで動作す

るプリンタユーザーインタフェースドライバ21で行う。以下、図1に示すシステムの動作を順に説明する。

【0066】まず、アプリケーション1は、任意の文書から成る印刷文書データを作成する。この印刷文書データは、既に説明したように、印刷文書メモリ24にファイル形式で格納される。次に、アプリケーション1を操作して印刷要求を行うと、プリンタユーザーインタフェースドライバ21は、所定の印刷条件設定用のダイアログを表示する。ここで、ユーザーはそのダイアログを見ながら、印刷ページ数、その他のプロパティを入力する。この処理はプロパティ設定処理部22によって周知の方法により行われる。

【0067】次に、ウォーターマークの重ね合わせ印刷を行おうとする場合、同様の画面中で、具体例1で説明したと同じように、ウォーターマークの指定を行う。図8には、プリンタユーザーインタフェースドライバのウォーターマーク指定のための動作フローチャートを示す。まず、ステップS1において、ウォーターマーク指定画面の表示を行う。ウォーターマークの文字の種類や大きさ等を選択するメニュー画面がここで表示される。これは具体例1の場合と同様である。ユーザーは画面に従いウォーターマークの形式を指定する（ステップS2）。

【0068】指定内容は自動的にチェックされ、指定エラーがあればステップS3からステップS2に戻り、形式指定を完了させる。形式指定が完了すると、ステップS3からステップS4に進み、ウォーターマーク情報をデバイス情報として保存する。即ち、ウォーターマークの具体的なプロパティに相当する情報をデバイス情報として、図1に示したデバイス情報メモリ25に格納する。具体的には、ウォーターマーク印刷データを作成するために必要な設定情報であるフォントやフォントサイズ等が、デバイス情報メモリ25中に格納される。

【0069】印刷要求に伴うプロパティの設定が終了すると、アプリケーション1は印刷開始を要求する。印刷開始によって、プリンタユーザーインタフェースドライバ21は、印刷文書メモリ24の印刷文書データファイルを開き、アプリケーション1を通じてスプーラ26にユーザーモードで動作する関数群を出力する。

【0070】図9に、以後の印刷データ生成動作のフローチャートを図示した。先ずステップS1において、アプリケーションで印刷実行プログラムを開始させる。関数群は、グラフィックデバイスインタフェース30に入力し、図9ステップS2、S3、S4に示したような関数を処理する。これらの処理は、デバイスコンテキスト作成処理のための予備処理である。次に、グラフィックデバイスインタフェース30は、スプーラ26から入力したデバイスに依存しないデータを、出力デバイスに依存するデータに変換する（ステップS5）。即ち、これから制御しようとするプリンタに適したコマンド群に変

換する。

【0071】次に、ステップS6において、印刷イベントのフックが行われる。即ち、通常は印刷動作が開始されると、グラフィックデバイスインタフェース30により作成された印刷データを、プリンタグラフィックドライバがビットマップデータに変換し、順にプリンタに転送するという動作を連続して行う。

【0072】しかしながら、ここでオブションとされたDrvDocumentEvent()関数をプログラム中に定義しておく、印刷イベントを一時的に停止し、ユーザーの指定した処理を挟むことができる。ここでは、グラフィックデバイスインタフェース30が、プリンタユーザーインタフェースドライバ21を呼び出し、プリンタユーザーインタフェースドライバ21に対しウォーターマークの生成処理を依頼する。即ち、ここで図1に示したウォーターマーク作成処理部23が呼び出され、動作を開始する。

【0073】ユーザーモードにおいては、Win32API（アプリケーションプログラムインタフェース）が利用できる。このWin32APIは、オペレーティングシステムに対し様々な処理を要求するコマンドで、このコマンドでウォーターマーク作成のためのプログラムを組み立てる。そのプログラムをプリンタユーザーインタフェースドライバ21のウォーターマーク作成処理部23で実行するようにしている。

【0074】図9のステップS7で、これからウォーターマークを付加する処理が必要になると判断すると、ステップS8に進む。プリンタユーザーインタフェースドライバ21は、デバイス情報メモリ25からデバイス情報を読み出して、その内容に合うウォーターマークのビットマップデータを作成する。即ち、プリンタグラフィックドライバ31が、ファイルを読み出す機能を持たないため、プリンタユーザーインタフェースドライバ21が、デバイス情報メモリ25からデバイス情報を読み出し、その内容を利用してウォーターマークのビットマップデータを作成する。

【0075】次に、プリンタユーザーインタフェースドライバ21は、カスタムエスケープ関数をコールする（ステップS9）。この関数によって、プリンタグラフィックドライバ31が呼び出される。プリンタグラフィックドライバ31は、カスタムエスケープ関数を受けて、ステップS10において、ウォーターマーク用のバッファを用意する。なお、ステップS10とステップS13に付けた\*印は、プリンタユーザーインタフェースドライバ21でなくプリンタグラフィックドライバが実行する処理であることを明確にするための印である。図の右側のフローチャートに示すその他の処理は、プリンタユーザーインタフェースドライバ側で実行する。

【0076】このステップS10の処理によって、図1に示したプリンタグラフィックドライバ31に所定のイ

メージバッファ33が設定される。そして、プリンタユーザーインタフェースドライバ21のウォーターマーク作成処理部23は、このイメージバッファ33に対しウォーターマークのビットマップデータを転送する（ステップS11）。次に、ステップS12において、再びカスタムエスケープ関数をコールする。

【0077】ここまでで、実質的にプリンタユーザーインタフェースドライバ21におけるウォーターマーク作成処理部23の処理は終了する。後はプリンタグラフィックドライバ31に処理が渡され、図9ステップS13のウォーターマーク印刷データと元データの論理和をとる処理が行われる。

【0078】即ち、図1に示したウォーターマーク重ね合わせ処理部34が起動し、イメージ生成処理部32で生成した印刷文書データのビットマップデータと、イメージバッファ33に格納したウォーターマークのビットマップデータとを重ね合わせて、これをデータ送信処理部35に転送する。なお、ウォーターマークは、ウォーターマークに相当する部分以外の部分は、透明になるようなデータとして処理される。両者を論理和演算することによって、ウォーターマークの上に印刷文書データが重ね合わされたイメージが得られる。

【0079】こうして、ステップS14、S15、S16の関数処理を行う。これらの処理は、デバイスコンテキストを作成するための後処理である。その後、デバイス送信処理部35の動作によって、作成されたビットマップデータが図示しないプリンタに転送され印刷される。

【0080】〈ウォーターマークの回転〉上記のような処理において、ウォーターマークに適当な傾斜を与えたい場合がある。これも、図1に示したプリンタユーザーインタフェースドライバ21のプロパティ設定処理22への設定により実行することができる。

【0081】図10には、回転させた文字をウォーターマークにするための処理フローチャートを示す。この処理は、図9のステップS8で実行する処理である。従って回転したウォーターマークのビットマップデータは、生成後そのままプリンタグラフィックドライバ31のイメージバッファ33に転送される。まず、ステップS1において、プリンタユーザーインタフェースドライバ21は、既にウォーターマーク設定処理によって作成されたデバイス情報を再びデバイス情報メモリ25から呼び出す。なお、このデバイス情報は、一般的な印刷のためのプロパティも含むデータである。ここからウォーターマーク情報を取り出す。そして、ステップS2において、フォントオブジェクトを作成する。即ち、ウォーターマークのための文字コードやフォントサイズ等の情報に基づいて、実際のフォントデータを作成する演算処理を行う。この演算処理は、ウォーターマークの実際のサイズを求めて、回転後のサイズ等を予測するためであ

る。

【0082】ステップS3では、文字列の大きさを算出する。ステップS4では、設定した回転角度で文字を回転させたときの描画領域を算出する。そして、ステップS5で、描画開始位置の算出を行う。ステップS3、S4、S5の処理について、これを具体的に図解したものを図11に示す。

【0083】図11には、ウォーターマーク文字列の回転による描画始点の算出説明図を示す。図には、用紙12のイメージとウォーターマーク13のイメージとが表示されている。この図に示すように、「文字列」というウォーターマークは、角度 $\theta$ で傾斜している。このようなウォーターマークの印刷を指定する場合には、予めウォーターマーク13のフォントを指定すると共に、その傾斜をプロパティ設定画面で指定する。

【0084】図10のステップS2で説明したフォントオブジェクトの作成処理では、この図面に示すように、ウォーターマーク13の実際のイメージを演算処理によって求める。その文字列の大きさは、図に示すように、高さがLL、幅がLWといった内容となる。このウォーターマーク13の描画範囲は、高さDL、幅DWの領域になる。DW、DL、LL、LWの関係は、簡単な三角関数によって図の式1、式2により求めることができる。また、描画開始点SP(X, Y)は、図の式3、式4によって求めることができる。

【0085】即ち、デバイスプロパティによってウォーターマークの傾きが設定されている場合、デバイス情報メモリ25からデバイス情報を読み出しながら、図11に示した演算処理を行うことによって、そのままウォーターマークを所定の傾きで描画することができる。こうして、描画したビットマップデータを、そのままスプーラ26を介してグラフィックデバイスインタフェース30に転送する。

【0086】〈具体例4の効果〉以上により、Windows NT4.0の場合であっても、ユーザーモードでウォーターマークのビットマップデータを自由に加工し、DrvDocumentEvent()関数によってプリンタグラフィックドライバ31とプリンタユーザーインタフェースドライバ21とが協調して、ウォーターマークのような地模様を印刷文書データに重ね合わせて、プリンタに供給することができる。

【図面の簡単な説明】

【図1】本発明による印刷システムのブロック図である。

【図2】プロパティ設定処理部の動作フローチャートである。

【図3】具体例1のプリンタドライバの動作フローチャートである。

【図4】具体例2のプリンタドライバの動作フローチャートである。

【図5】ウォーターマークの伸縮処理動作説明図である。

【図6】白抜き処理の動作説明図である。

【図7】具体例4の印刷システムブロック図である。

【図8】プリンタユーザインタフェースドライバの動作フローチャートである。

【図9】印刷データ生成動作のフローチャートである。

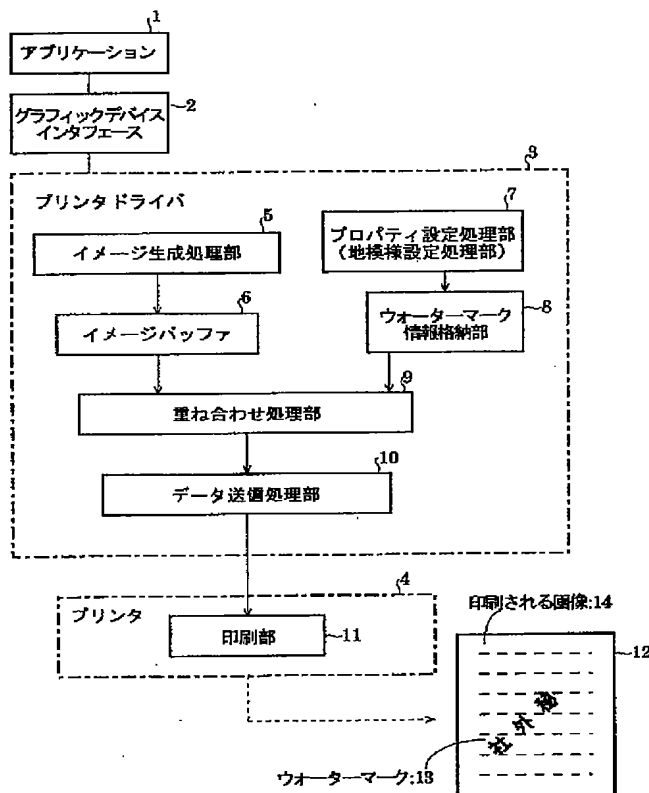
【図10】回転させた文字をウォーターマークにするための処理フローチャートである。

【図11】ウォーターマーク文字列の回転による描画始点の算出動作説明図である。

【符号の説明】

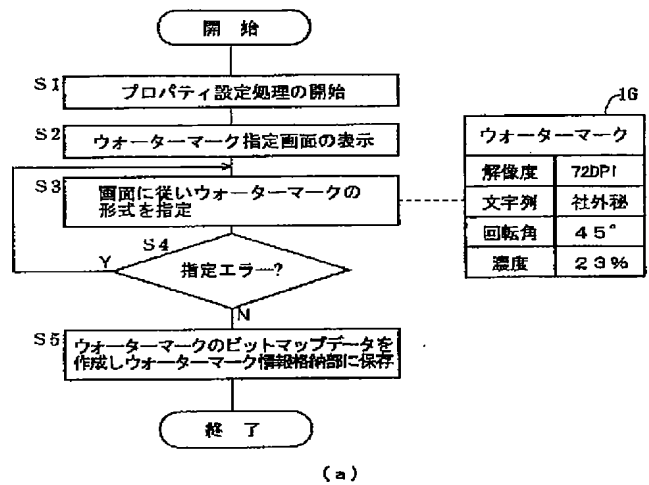
- 1 アプリケーション
- 2 グラフィックデバイスインタフェース
- 3 プリンタドライバ
- 4 プリンタ
- 5 イメージ生成処理部
- 7 プロパティ設定処理部
- 9 重ね合わせ処理部
- 10 データ送信処理部
- 12 用紙
- 13 ウォーターマーク
- 14 印刷される画像

【図1】

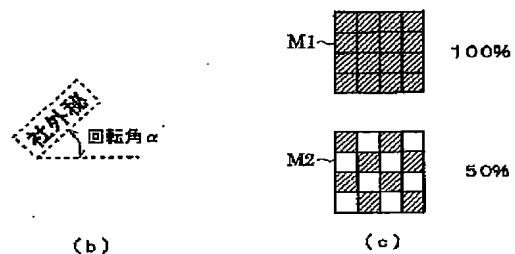


本発明による印刷システムのブロック図

【図2】



(a)

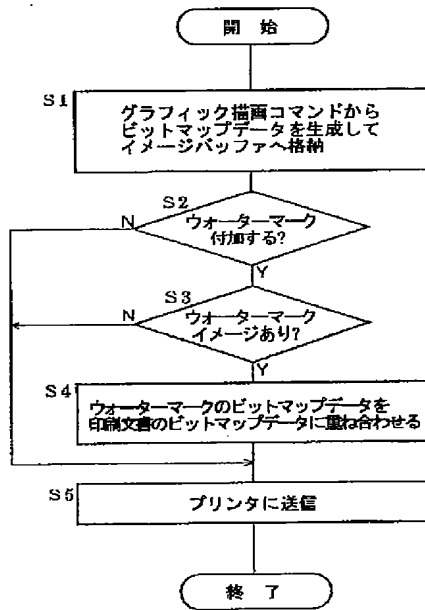


(b)

(c)

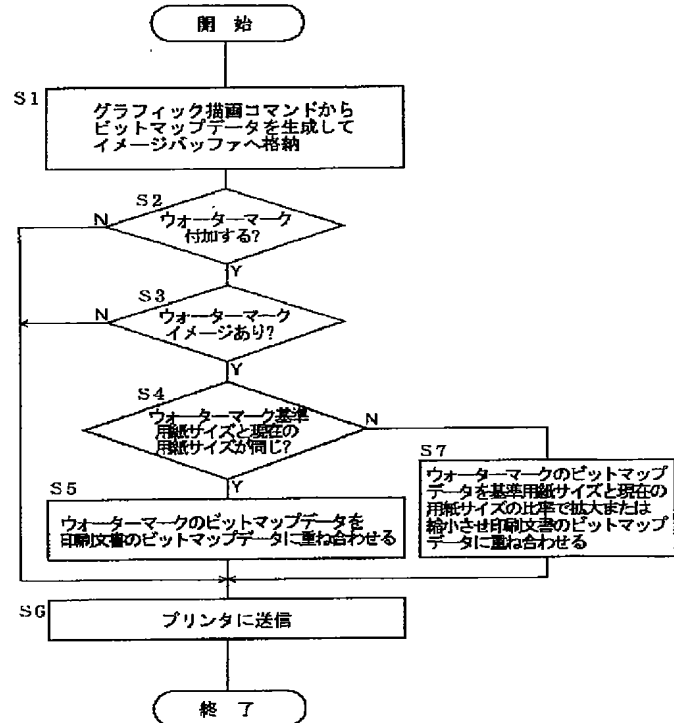
プロパティ設定処理部の動作フローチャート

【図 3】



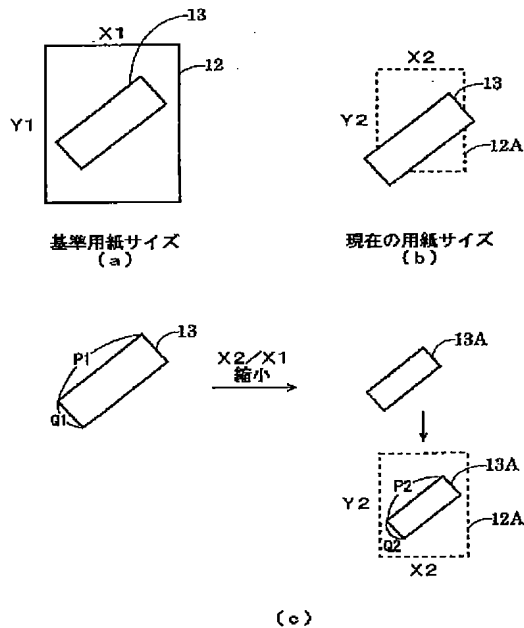
具体例 1 のプリンタドライバ動作

【図 4】



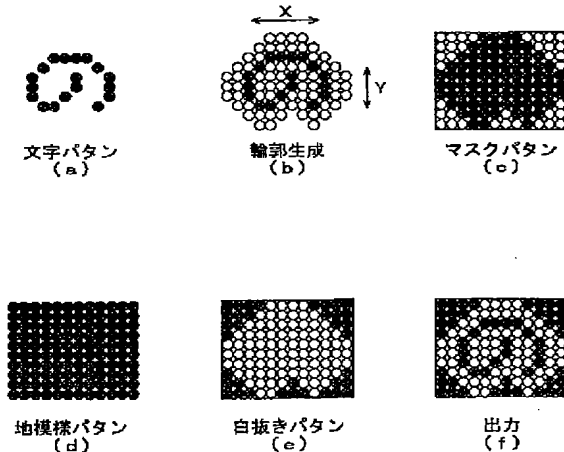
具体例 2 のプリンタドライバ動作

【図 5】



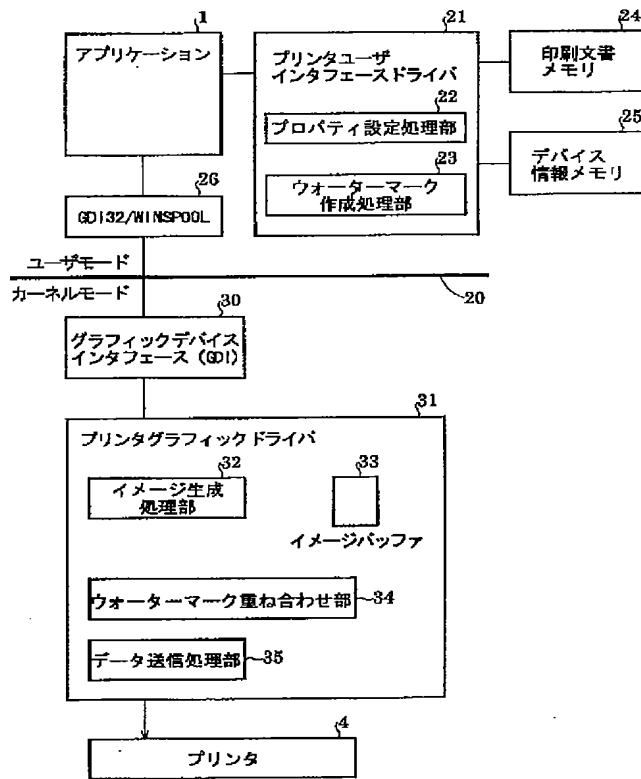
ウォーターマークの伸縮処理動作説明図

【図 6】



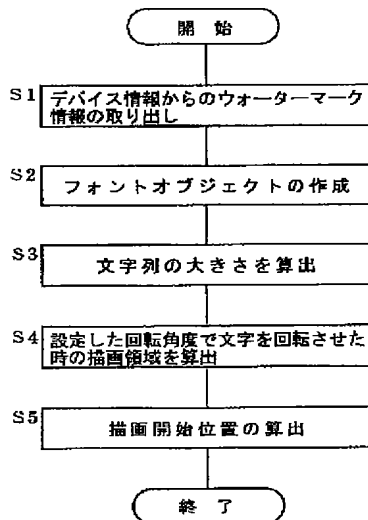
白抜き処理の動作説明図

【図 7】



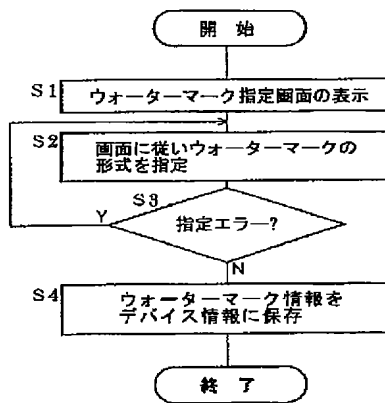
具体例4の印刷システムブロック図

【図 10】



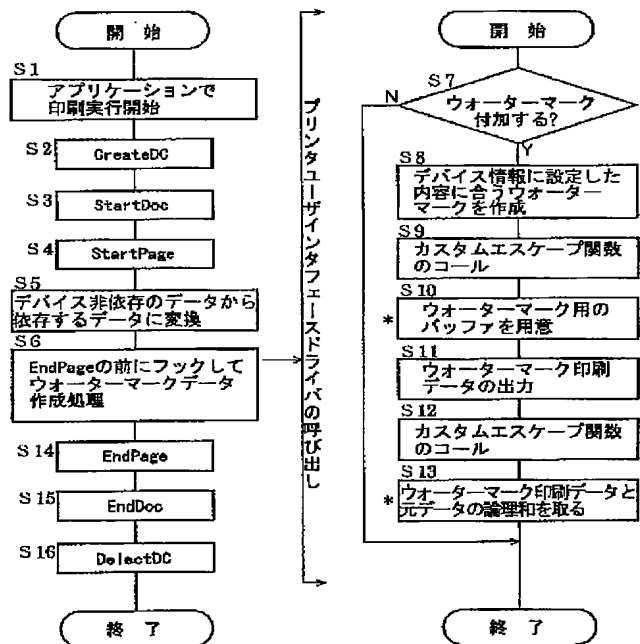
回転させた文字をウォーターマークにするための処理

【図 8】



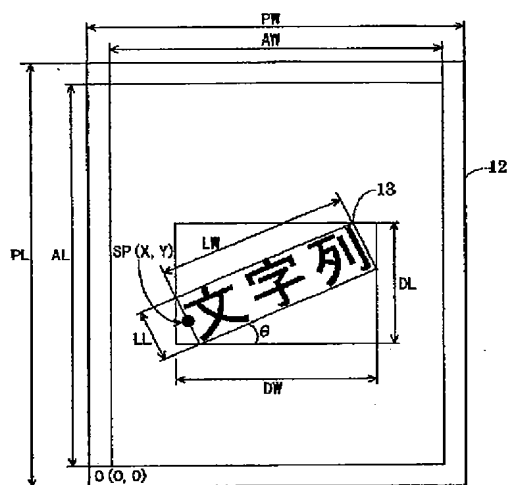
プリンタユーザインタフェースドライバの動作

【図 9】



印刷データ生成動作のフローチャート

【図 1 1】



PL: 用紙長  
 PW: 用紙幅  
 AL: 印刷可能範囲高さ  
 AW: 印刷可能範囲幅  
 θ: 回転角度  
 LL: 文字列の高さ  
 LW: 文字列の幅  
 DL: 文字列の描画範囲高さ  
 DW: 文字列の描画範囲幅  
 SP(X, Y): 描画の始点 (X: X座標、Y: Y座標)

$$DW = LW * \cos \theta + LL * \sin \theta \quad \cdots \text{式 1}$$

$$DL = LW * \sin \theta + LL * \cos \theta \quad \cdots \text{式 2}$$

$$X = (AW - DW) / 2 + LL * \sin \theta / 2 \quad \cdots \text{式 3}$$

$$Y = (AL - DL) / 2 + LL * \cos \theta / 2 \quad \cdots \text{式 4}$$

ウォーターマーク文字列の回転による描画始点の算出動作